

学修番号 17890543

修士論文

目的言語の文書文脈を用いたニューラル機械翻訳

山岸 駿秀

2019 年 2 月 22 日

首都大学東京大学院
システムデザイン研究科 情報通信システム学域

山岸 駿秀

審査委員：

小町 守 准教授 （主指導教員）

山口 亨 教授 （副指導教員）

高間 康史 教授 （副指導教員）

目的言語の文書文脈を用いたニューラル機械翻訳*

山岸 駿秀

修論要旨

ある言語で書かれた文を別の言語で書かれた文へ変換する技術である機械翻訳は、様々な国や地域の人が円滑なコミュニケーションを行う上で重要な技術である。しかしこれまで発表された機械翻訳システムの多くは、問題を簡略化するため 1 文単位の入出力を前提にしている。文書を訳したい場合であっても 1 文ずつ独立に翻訳を行う必要があるため、最終的に得られた文書には訳語や文体等の一貫性があることが保証されない。これは、機械翻訳と翻訳者による翻訳との大きな違いの 1 つである。したがって、機械翻訳に文書の文脈を取り込むことは、機械翻訳の性能が人間の翻訳者のレベルに到達するために重要なステップであると考えられる。

文脈の必要性は、言語対によっても変化する。例えば英独翻訳では、ドイツ語の三人称単数の代名詞は名詞ごとに定められた文法上の性によって変化するため、場合によっては文脈を参照しなければならない。日英翻訳においては、日本語では文脈に存在する情報については全て省略できる一方、英語では主語を必須とするため、より流暢な訳出を行うために文脈が必要になる。一般的に、別語族の言語間の翻訳時は文書構造が異なる可能性が高いため、文脈の必要性が増すと考えられる。

近年の機械翻訳の研究開発では、従来の機械翻訳と比較して流暢性の高い文を生成できることから、深層ニューラルネットワークを用いたニューラル機械翻訳と呼ばれる手法が主流となっている。ニューラル機械翻訳は、入力文を読み込むための Encoder 機構、Encoder で得られた情報から適宜必要な情報を参照するための Attention 機構、Encoder と Attention で得られた情報から文を生成するための Decoder 機構の 3 つからなる。ニューラル機械翻訳は、原理的には入力文中の全ての単語間の関係を捉えられるとされており、この捉えられる関係の幅を単語間から文間へと拡張し、文脈つきニューラル機械翻訳へと発展させようとする動きがある。

*首都大学東京大学院 システムデザイン研究科 情報通信システム学域 修士論文, 学修番号 17890543, 2019 年 2 月 22 日.

本研究は、目的言語側の文脈を取り入れることによって文脈つきニューラル機械翻訳の性能を向上させることを目的とする。Multi-Encoder で 1 つ前の出力文が有用でないのは、それが入力文を扱う機構である Encoder で処理されていることが原因であると考え、出力文を扱う機構である Decoder で処理される必要があるという仮説を立てた。これを実証するために、1 つ前の文を翻訳した際の Decoder の状態を全て保存しておき、対象文の翻訳時に保存された状態ベクトル集合に対して Attention を適用する手法を提案する。これにより、Decoder を通して目的言語側の文脈を扱うことができると考えた。また、この提案手法は計算済みのベクトル列を用いるため、先行研究と比較してメモリ使用量や計算時間等を削減することができる。本論文では、この手法について、日英・英日、独英・英独、中英・英中の 6 言語対で実験を行った。この結果、英独・独英、英中・中英の実験では原言語側より目的言語側の文脈が有用であることや、日英・英日の言語対ではどちら側の文脈も同等に有用であることなどがわかった。

本研究の主な貢献は以下の 3 点である。

- 目的言語側の文脈をより適切に扱う手法を提案した。
- 原言語側の文脈と目的言語側の文脈のどちらがより有用であるかを検証するために複数の言語対で実験を行い、言語対に大きく依存するという結論を示した。
- 文脈つきニューラル機械翻訳に必要なメモリ使用量や計算時間を削減できる。

本論文の構成は次の通りである。第 1 章で概要や主要な貢献等について述べる。第 2 章ではニューラル機械翻訳の仕組みと近年提案された文脈つきニューラル機械翻訳の先行研究について概説する。第 3 章では本研究の提案手法を説明する。第 4 章では実験設定や実験によって得られた結果について述べる。第 5 章では結果についての考察を行い、第 6 章では本研究のまとめと今後の課題について述べる。

Neural Machine Translation with Target-side Context*

Hayahide Yamagishi

Abstract

Neural machine translation has become popular in recent years because it can handle larger contexts compared to conventional machine translation systems. However, most of neural machine translation architectures do not employ document-level contexts due to task complication.

Recently, several studies on neural machine translation have attempted to use document-level context by using a multi-encoder and two attention mechanisms to read the current and previous sentences to incorporate the context of previous sentences. These studies concluded that the target-side context is less useful than the source-side one. However, we hypothesized that the reason why the target-side context is less useful lies in the architecture to model context.

In this study, we investigate how the target-side context can improve context-aware neural machine translation. We propose a weight sharing method in which neural machine translation saves decoder states and calculates an attention vector using the saved states when translating a current sentence. Our experiments show that target-side context is more useful than source-side context if we plug it into neural machine translation as decoder states when translating a previous sentence.

*Master's Thesis, Department of Information and Communication Systems, Graduate School of System Design, Tokyo Metropolitan University, Student ID 17890543, February 22, 2019.

目次

図目次		vi
表目次		vii
第 1 章	はじめに	1
第 2 章	先行研究	4
2.1	ニューラル機械翻訳の概説	4
2.2	文脈つきニューラル機械翻訳の先行研究	6
2.2.1	Multi-Encoder 型の先行研究	6
2.2.2	その他	7
第 3 章	Multi-Encoder を用いた文脈つきニューラル機械翻訳	9
3.1	Separated 型	9
3.2	Shared 型	11
第 4 章	実験	13
4.1	データ	13
4.2	ニューラル機械翻訳の設定	14
4.3	結果	16
第 5 章	考察	17
5.1	言語対の影響	17
5.2	学習の収束	18

5.3	重み共有の影響	18
5.4	出力例	19
第 6 章	おわりに	23
	発表リスト	24
	謝辞	26
	参考文献	27

図目次

1.1	用語の定義	2
2.1	Luong 型の Attention を備えたニューラル機械翻訳モデル	5
3.1	Separated 型の Multi-Encoder モデル	10
3.2	Shared 型の Multi-Encoder モデル	11
3.3	Shared Mix 型の Multi-Encoder モデル	12
5.1	各実験の学習時における検証データの BLEU スコアの変化	21

表目次

4.1	前処理を通して得られた各コーパスの統計量	14
4.2	各実験の BLEU スコア	15
5.1	Recipe 日英翻訳の出力例	22

第 1 章 はじめに

ある言語で書かれた文を別の言語で書かれた文へ変換する技術である機械翻訳は、様々な国や地域の人が互いに円滑なコミュニケーションを行う上で重要な技術である。これまで発表された機械翻訳システムのほとんどは、問題やその処理を簡略化するために 1 文単位の入出力を前提としている。数文からなる文書を訳したい場合であっても 1 文ずつ独立に翻訳を行う必要があるため、最終的に得られた文書には訳語や文体等の一貫性があることが保証されない。一方で人間の翻訳者が行う翻訳では、文書全体を俯瞰することで文脈を扱えるため、全体の文体や流れを保ったまま翻訳することができる。これは、現在の機械翻訳による生成と翻訳者による訳出との間にある大きな違いの 1 つである。したがって、機械翻訳に文脈の情報を取り込むことは、機械翻訳の性能が人間の翻訳者のレベルに到達するために重要なステップであると考えられる。

自然言語処理では一般に、文中のある単語に着目したときにその周辺に存在する単語を文脈語、または単に文脈 (context) という。一方、本論文で取り扱う文脈はそれとは異なり、ある文に着目したときにその文より前の文で述べられた話題一般のことを指している。本論文で使用する用語について、以下のように定義する。また、図 1.1 に例を挙げる。

対象文 ある時刻 i におけるシステムの入力文または出力文。

文脈文 時刻 $i - 1$ におけるシステムの入力文または出力文。

文脈 時刻 1 から時刻 $i - 1$ までの文脈文で言及された全ての情報。

文脈の必要性は、言語対によっても変化する。例えば英独翻訳では、ドイツ語の三人称単数の代名詞は名詞ごとに定められた文法上の性によって変化するため、場合によっては文脈文を参照しなければならない。日英翻訳においては、日本語では文脈に存在する情報については全て省略できる一方、英語では主語を必須とするため、より流暢な訳出を行うために文脈が必要になる。一般的に、異なる語族の言語であるなど、言語の類似性が小さい言語対の翻訳時は互いの持つ文書構造に違いがある可能性が高いため、文脈の必要性が増すと考えられる。

近年の機械翻訳の研究開発では、他の手法と比較して流暢性の高い文を生成でき

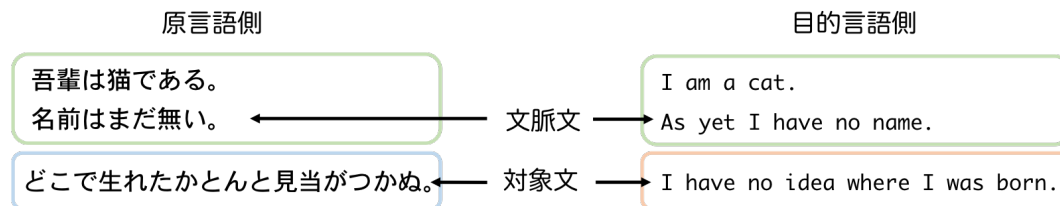


図 1.1: 用語の定義

ることから，深層ニューラルネットワークを用いたニューラル機械翻訳（**Neural Machine Translation, NMT**）[1] と呼ばれる手法が主流となっている．深層ニューラルネットワークには，主に Recurrent Neural Network（RNN）が用いられる [2, 3] が，Convolutional Neural Network（CNN）を用いる ConvS2S という手法 [4] や多層の Feedforward Neural Network（FFNN）を用いる Transformer という手法 [5] など提案されており，特に Transformer は現在の機械翻訳研究において数多くの言語対の最高性能を達成している．現在のニューラル機械翻訳は一般的に，入力文を読み込むための **Encoder** 機構，Encoder で得られた情報から適宜必要な情報を参照するための **Attention** 機構，Encoder と Attention から与えられた情報をもとに文を生成するための **Decoder** 機構の 3 つの構造を持つ．これらの機構の詳細については 2.1 節で述べる．

ニューラル機械翻訳は，原理的には入力文中の全ての単語間の関係を捉えられるとされているため，この捉えられる関係の幅を単語間から文間へと拡張し，文書単位の機械翻訳（document-level machine translation）を目指す動きがある．しかし，先行研究のほとんどは文脈の情報を部分的に取り入れることに留めており，これらは文脈つき機械翻訳（**context-aware machine translation**）と呼ばれて区別されている．本研究の提案手法も文脈つき機械翻訳に分類される．

文脈つき機械翻訳をニューラル機械翻訳の枠組みで構築する文脈つきニューラル機械翻訳では，Multi-Encoder と呼ばれる機構を用いることが一般的である．これはニューラル機械翻訳に備わっている Encoder に加え，文脈文を読み込むための Encoder とその情報を参照するための Attention を追加するものである．先行研究では Multi-Encoder で原言語側の文脈文を読み込むことで性能が向上したが，目的言語側の文脈文を読み込んだ場合に性能が向上しないとの報告があった．したがっ

て現在の枠組みでは、原言語側の文脈を使うことで性能を上げられる一方、目的言語側の文脈は性能向上に寄与しないと結論づけられている。

本研究は、目的言語側の文脈を取り入れることによって文脈つきニューラル機械翻訳の性能を向上させることを目的とする。Multi-Encoder で目的言語側の文脈文が有用でないのは、それが入力文を扱う機構である Encoder で処理されていることが原因であると考え、出力文を扱う機構である Decoder で処理される必要があるという仮説を立てた。これを実証するために、文脈文を翻訳した際の Decoder の状態を全て保存しておき、対象文の翻訳時に保存された状態ベクトル集合に対して Attention を適用する手法を提案する。これにより、Decoder を通して目的言語側の文脈を扱うことができると考えた。また、この提案手法は計算済みのベクトル列を用いるため、先行研究と比較してメモリ使用量や計算時間等を削減することができる。本論文では、この手法について、日英・英日、独英・英独、中英・英中の 6 言語対で実験を行った。この結果、英独・独英、英中・中英の実験では原言語側より目的言語側の文脈が有用であることや、日英・英日の言語対ではどちら側の文脈も同等に有用であることなどがわかった。

本研究の主な貢献は以下の 3 点である。

- 目的言語側の文脈をより適切に扱う手法を提案した。
- 原言語側の文脈と目的言語側の文脈のどちらがより有用であるかを検証するために複数の言語対で実験を行い、言語対に大きく依存するという結論を示した。
- 文脈つきニューラル機械翻訳に必要なメモリ使用量や計算時間を削減できる。

本論文のこれ以降の構成は次の通りである。第 2 章ではニューラル機械翻訳の仕組みと近年提案された文脈つきニューラル機械翻訳の先行研究について概説する。第 3 章では本研究の提案手法を説明する。第 4 章では実験設定や実験によって得られた結果について述べる。第 5 章では結果の考察を行い、第 6 章では本研究のまとめを述べる。

第 2 章 先行研究

本章ではニューラル機械翻訳の概説とその発展である文脈つきニューラル機械翻訳の概説を行う。

2.1 ニューラル機械翻訳の概説

2014 年に Sutskever ら [1] が提案した Recurrent Neural Network (RNN) を用いたニューラル機械翻訳は、現在の機械翻訳技術の基礎となっている。現在の機械翻訳は、入力文の情報を適宜参照することができる、Attention 機構を用いたものが一般的である。Attention 機構には大きく分けて Bahdanau 型 [2] と Luong 型 [3] の 2 つがある。本研究は Luong 型を拡張したものであるため、本節ではこの概略を述べる。

図 2.1 に概略図を示す。ニューラル機械翻訳モデルは、入力文を読み込んで各時刻（各単語）の隠れ層ベクトルを計算する Encoder 機構（図中の青部）、入力文のどこに着目して出力するかを決める Attention 機構（図中の緑部）、Attention の情報と過去の出力履歴から各時刻において適切な単語を生成する Decoder 機構（図中の橙部）の 3 つからなる。入力文 X を単語 x 、文長 M の列 $X = x_1, \dots, x_m, \dots, x_M$ 、出力文 Y を単語 y 、文長 N の列 $Y = y_1, \dots, y_n, \dots, y_N$ とする。目的関数は以下の式で与えられ、確率 $p(Y|X)$ を最大化するように学習を行う。

$$p(Y|X) = \prod_{n=1}^N p(y_n|y_{<n}, X) \quad (2.11)$$

以下の式で Encoder の隠れ層ベクトル s_t を計算する。

$$s_m = \text{LSTM}_{\text{enc}}(e_{x,m}, s_{m-1}) \quad (2.12)$$

$$e_{x,m} = W_x x_m \quad (2.13)$$

ここで、LSTM とは Long Short-Term Memory[6] を示す。 $W_x \in \mathbb{R}^{E \times V}$ は単語を受け取って単語分散表現を返す重み行列*である。Decoder は以下の式に従って

* E は単語分散表現の次元数、 V は語彙空間の大きさを示す。

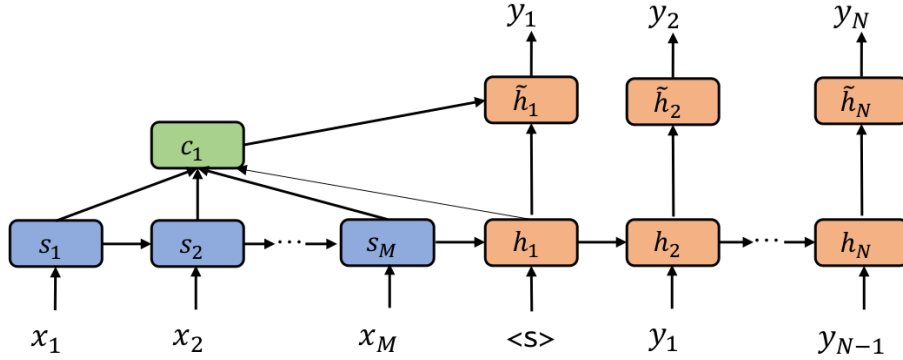


図 2.1: Luong 型の Attention を備えたニューラル機械翻訳モデル

単語を生成する.

$$p(y_n|y_{<n}, X) = \text{softmax}(W_o \tilde{\mathbf{h}}_n) \quad (2.14)$$

$$\tilde{\mathbf{h}}_n = W_h[\mathbf{h}_n; \mathbf{c}_n] \quad (2.15)$$

$$\mathbf{h}_n = \text{LSTM}_{\text{dec}}(\mathbf{e}_{y,n}, \mathbf{h}_{n-1}) \quad (2.16)$$

$$\mathbf{e}_{y,n} = W_y y_{n-1} \quad (2.17)$$

ここで、記号 $[\mathbf{a}; \mathbf{b}]$ はベクトル \mathbf{a} とベクトル \mathbf{b} の結合 (concatenation) を指し、 \mathbf{a} , \mathbf{b} をそれぞれ n 次元ベクトルとすると、 $[\mathbf{a}; \mathbf{b}]$ は $2n$ 次元ベクトルとなる。 $W_h \in \mathbb{R}^{H \times 2H}$ は重み行列[†]であり、 $W_y \in \mathbb{R}^{E \times V}$ は W_x と同様に単語分散表現を扱う重み行列である。 \mathbf{c}_n は Attention ベクトルであり、式 2.18 で得られる。なお、図 2.1 では簡略化のために y_1 の計算時のみ Attention ベクトルが用いられているように表現したが、実際の処理中は各時刻 n において Attention ベクトルを計算する。

$$\mathbf{c}_n = \sum_{m=1}^M \alpha_{n,m} \mathbf{s}_m \quad (2.18)$$

$$\alpha_{n,m} = \text{softmax}(\mathbf{s}_m \cdot \mathbf{h}_n) \quad (2.19)$$

Attention ベクトルは、各時刻 n の翻訳時に入力文の特定の単語の情報をを用いるために計算される。

[†] H は隠れ状態ベクトルの次元数を示す。

2.2 文脈つきニューラル機械翻訳の先行研究

Müller ら [7] によると，現在の文脈つきニューラル機械翻訳の研究は，文脈文を取り込むための Encoder（文脈 Encoder）を追加した Multi-Encoder 型と，それ以外の様々なネットワークを用いたモデルの 2 つに大別される．本章ではこの分類法に従い，文脈つきニューラル機械翻訳の先行研究を概説する．

2.2.1 Multi-Encoder 型の先行研究

Tiedemann と Scherrer [8] は，従来のニューラル機械翻訳の枠組みで，以下のよう
に 2 文を 1 文とみなして扱うことで文脈文を導入した．

I am a cat . <CONCAT> As yet I have no name .

2 文を 1 文にまとめた入力文から 2 文を 1 文にまとめた出力文に翻訳する “2to2”
や，同様の入力文から 1 文へ翻訳する “2to1” という手法を提案した．英独翻訳で
実験したところ，2to2 で性能向上が認められた．この研究は，文脈 Encoder の数
が 0 の場合の Multi-Encoder 型とみなせる．著者らは，2to2 で性能が向上した理
由として，目的言語側の文脈を部分的に取り込んだことを挙げている．本研究でも
目的言語側の文脈を用いることで性能向上が認められており，この研究の知見をさ
らに広げることができた．

Jean ら [9] は Multi-Encoder による文脈つきニューラル機械翻訳を初めて提案し
た．この論文では詳細な計算式等については示されていないが，英独翻訳において
原言語側の文脈文を用いることで性能向上が認められたという．Bawden ら [10] は
Jean らの研究をまとめ，2to2 や 2to1 と Multi-Encoder 型の性能を自身の作成し
た評価データを用いて比較した．英仏翻訳で実験したところ，2to2 や原言語側の文
脈文を扱う Multi-Encoder の性能は向上したが，目的言語側を扱う Multi-Encoder
の性能が低かった．本研究は Bawden らの研究を元にしており，Multi-Encoder 型
での目的言語側の文脈文の扱い方を提案するものである．

Voita ら [11] は Transformer [5] を用いた Multi-Encoder 型を提案した．原言
語側の文脈文を使うモデルを英露翻訳で実験したところ，わずかに性能が向上し
た．また，Attention ベクトル列を可視化し，このモデルが共参照解析を解いてい

ると考えられるような手がかりを見つけた．ここで共参照解析とは，文章内で同一の実態を指す代名詞と名詞（句）が同じクラスに属するようにクラスタリングするタスクである．このモデルを共参照解析器とみなして評価すると，一般的に使われているツールと同程度の性能であった．したがって，Multi-Encoder 型の文脈つきニューラル機械翻訳は代名詞の同定を行えていると思われる．Zhang ら [12] も同時期に同様の手法を提案している．中英翻訳で実験を行い，自動評価だけでなく人手評価のスコアも向上することを示した．また，この論文では事前に文脈を用いないニューラル機械翻訳を学習しておき，得られた重みを文脈つきニューラル機械翻訳の初期値として与えることが有用であると示した．これらの研究では目的言語側の文脈文の有用性について評価していないため，本研究ではそれに取り組む．本論文で行う実験では，Zhang らの提案した事前学習手法を用いる．

Müller ら [7] は，それまでに提案された文脈つきニューラル機械翻訳の手法を Multi-Encoder 型とその他に分類し，前者の性能を著者らの作成した英独翻訳のデータセットで比較した．さらに，文脈 Encoder の単語分散表現を Encoder あるいは Decoder のものと共有する手法を提案し，わずかに性能が向上することを示した．本研究で述べる Shared 型はこの共有手法をさらに拡張し，文脈 Encoder の全ての重みを共有するものである．

2.2.2 その他

Wang ら [13] は，単語ベクトルから文ベクトルを作る Encoder と，文ベクトルから文書ベクトルを作る Encoder の 2 層からなるモデルを提案した．階層的にすることで容易に複数文を扱うことが可能である．しかし，階層型にすることによってモデルのパラメータ数や計算コストが大きくなることや，目的言語側の文脈を考慮できない点などにおいて，本研究の提案手法とは異なっている．

Tu ら [14] は，翻訳時の Attention ベクトルの履歴をキャッシュで管理することにより，適切なタイミングで適切な文脈を使える手法を提案した．著者らは，キャッシュについての分析を行い，概ね 5 文より前の翻訳履歴については性能向上につながらないことを示した．しかし，この手法は生成された単語を用いてキャッシュを管理するため，生成を間違えた場合にその間違いが悪影響を与える可能性が

ある．本研究で提案する Shared 型は生成された単語を必要としないため，そのような影響を小さくすることができる．

Maruf ら [15] は，Memory Network を用いた手法を提案した．著者らは，実験結果から目的言語側の文脈は原言語側の文脈と同程度かそれ以上に重要であることを示している．本研究でも様々な言語対についての実験を行い，Multi-Encoder 型でも同様の知見を得た．しかし，Maruf らの手法ではニューラルネットワークの構造が大規模になることから，計算資源等の制限がかかる．本論文の提案手法は，先行研究と比較して文脈つきニューラル機械翻訳をととてもコンパクトに実現できる可能性を提示している．

第3章 Multi-Encoder を用いた文脈つきニューラル機械翻訳

本研究のタスクは、 L 文対を含む文書を $D = (X^1, Y^1), \dots, (X^i, Y^i), \dots, (X^L, Y^L)$ としたときに各文 X^i に対応する Y^i を得ることである．ここで、 X^i と Y^i はそれぞれ原言語側の文と目的言語側の文を示す． X^i は文長 M^i と単語 x_m^i を用いて $X^i = x_1^i, \dots, x_m^i, \dots, x_{M^i}^i$ とする． Y^i についても同様に、文長 N^i と単語 y_n^i を用いて $Y^i = y_1^i, \dots, y_n^i, \dots, y_{N^i}^i$ とする．

本研究では、Multi-Encoder の枠組みで目的言語側の文脈を扱うための手法を提案する．本章では、先行研究で提案された手法を改変した Separated 型と提案手法である Shared 型についてそれぞれ述べる．

3.1 Separated 型

Multi-Encoder を用いた文脈つきニューラル機械翻訳モデルの目的関数を式 3.11 で定義する．ただし、 Z^{i-1} は文脈文を示し、 X^{i-1} または Y^{i-1} が入るものとする．

$$p(Y^i | X^i, Z^{i-1}) = \prod_{n=1}^{N^i} p(y_n^i | y_{<n}^i, X^i, Z^{i-1}) \quad (3.11)$$

$$p(y_n^i | y_{<n}^i, X^i, Z^{i-1}) = \text{softmax}(W_o \tilde{h}_n^i) \quad (3.12)$$

Bawden ら [10] の手法では \tilde{h}_n^i を以下のように計算する．隠れ状態ベクトルのサイズを H とすると $W_h, W_c \in \mathbb{R}^{H \times 2H}$ であるから、この部分のパラメータ数は $4H^2$ である．

$$\tilde{h}_n^i = W_h[h_n^i; a_n^i] \quad (3.13)$$

$$a_n^i = W_c[c_n^i; c_n^{i-1}] \quad (3.14)$$

しかし本研究では、パラメータ数の削減や分析の容易さなどから、式 3.15 で計算する．このときのパラメータ数は $W_h \in \mathbb{R}^{H \times 3H}$ より、 $3H^2$ である．

$$\tilde{h}_n^i = W_h[h_n^i; c_n^i; c_n^{i-1}] \quad (3.15)$$

式 3.15 以降の計算は、2.1 節で示した従来のニューラル機械翻訳と同様である．

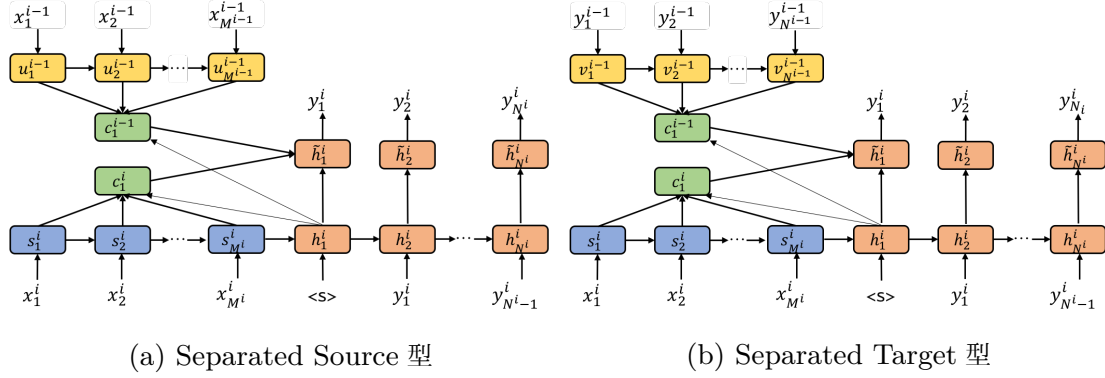


図 3.1: Separated 型の Multi-Encoder モデル

式 3.15 と式 2.15 を比較すると、 \mathbf{c}_n^{i-1} が増えていることがわかる．これは文脈文の情報に対して計算した Attention（文脈 Attention）であり，文脈文のどの単語に注目して生成するかを決めるベクトルである．文脈 Attention \mathbf{c}_n^{i-1} は以下の式 3.16 で得られる．

$$\mathbf{c}_n^{i-1} = \sum_{t=1}^{|Z^{i-1}|} \beta_{n,t} \mathbf{z}_t^{i-1} \quad (3.16)$$

$$\beta_{n,t} = \text{softmax}(\mathbf{z}_t^{i-1} \cdot \mathbf{h}_n^i) \quad (3.17)$$

ここまでの構造は本節で説明する Separated 型と次節で述べる Shared 型に共通であり，異なる点は式 3.16 および式 3.17 で用いる \mathbf{z} の算出方法にある．

Separated 型の概略を図 3.1 に示す．Separated 型では，文脈文 Z^{i-1} の情報 \mathbf{z}^{i-1} を得るために文脈 Encoder（図中の黄部）を導入する．ここで， $\mathbf{z}^{i-1} = \mathbf{u}^{i-1}$ または \mathbf{v}^{i-1} とし， Z^{i-1} が原言語側の文脈文 X^{i-1} の場合は \mathbf{u}^{i-1} ， Z^{i-1} が目的言語側の文脈文 Y^{i-1} の場合は \mathbf{v}^{i-1} が得られるとする．

$$\mathbf{u}_t^{i-1} = \text{LSTM}_{\text{src_enc}}(W_x x_t^{i-1}, \mathbf{u}_{t-1}^{i-1}) \quad (3.18)$$

$$\mathbf{v}_t^{i-1} = \text{LSTM}_{\text{trg_enc}}(W_y y_t^{i-1}, \mathbf{v}_{t-1}^{i-1}) \quad (3.19)$$

この提案手法は，Encoder と文脈 Encoder が分かれているため，Separated 型と呼ぶ．さらに，原言語の文脈文を読み込む場合を Separated Source 型，目的言語

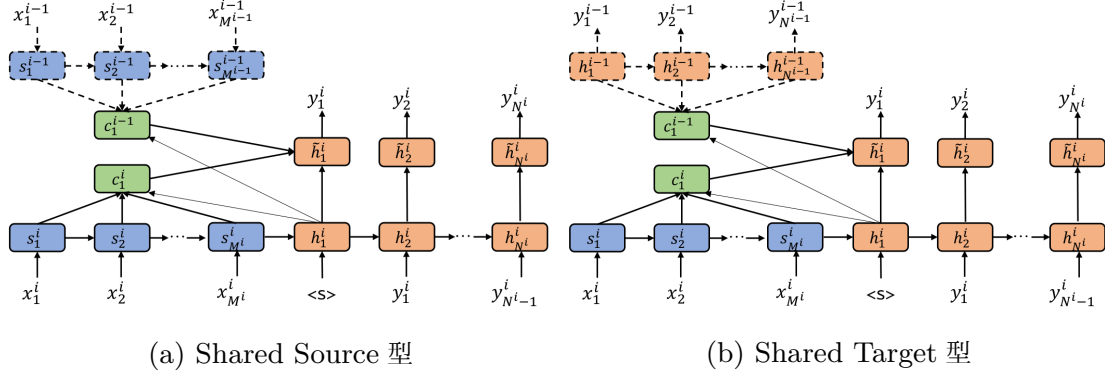


図 3.2: Shared 型の Multi-Encoder モデル

側の文脈文を読み込む場合を Separated Target 型と呼ぶことにする．Separated Source 型はこれまで提案されてきた Multi-Encoder 型の手法と同様のものとして扱うが，Separated Target 型は出力側となる目的言語の文脈文を Encoder の枠組みで扱うことが特徴である．

3.2 Shared 型

Shared 型の概略を図 3.2 に示す．Separated 型と同様，原言語側の文脈文を用いる場合は Shared Source 型，目的言語側の文脈文を用いる場合は Shared Target 型とする．この手法では文脈文の翻訳時に計算した状態ベクトルをそのまま用いる．したがって，Shared Source 型では $z = s^{i-1}$ ，Shared Target 型では $z = h^{i-1}$ である．これを用いて式 3.16 と式 3.17 を計算し，文脈 Attention c_n^{i-1} を得る．Separated 型とは異なり，LSTM を用いた文脈 Encoder を持たないが，仮想的に文脈 Encoder を持つと考えてこちらの機構についても文脈 Encoder と呼ぶ．図中では，学習時に誤差が伝播しない箇所を点線で記した．Shared 型では文脈 Encoder の学習を行っていないことがわかる．

このモデルは，文脈 Encoder と Encoder または Decoder がニューラルネットワークの重み行列を共有しているとみなせる．特に，目的言語側の文脈文を扱う場合 (Shared Target 型) は，Encoder の重み行列ではなく Decoder の重み行列を共

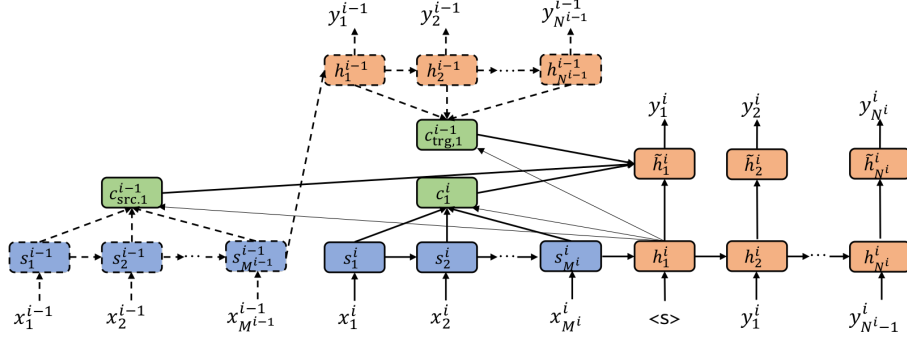


図 3.3: Shared Mix 型の Multi-Encoder モデル

有している． h^{i-1} は Decoder の状態ベクトルであるため，目的言語側の文脈文を Decoder を通して扱うことができる．

原言語側の文脈文を扱う場合であっても，重み行列を共有することでパラメータ数を削減できる．また，状態ベクトルを得るために追加の計算を必要としないため，Separated 型と比較して計算時間やメモリ使用量等を削減することができる．したがってこの提案手法を使うことで，従来の文脈つきニューラル機械翻訳をコンパクトにすることができる．

Shared 型に限り，両方の文脈文を用いた Shared Mix 型の実験を行った．図 3.3 に概略を示す．このモデルは Shared Source 型で得られた文脈 Attention $c_{src,n}^{i-1}$ と Shared Target 型で得られた文脈 Attention $c_{trg,n}^{i-1}$ の和を \tilde{h}_n^i の計算に用いる．

$$\tilde{h}_n^i = W_h[h_n^i; c_n^i; c_{src,n}^{i-1} + c_{trg,n}^{i-1}] \quad (3.21)$$

複数の状態ベクトルや Attention ベクトルをまとめる方法は，主に和をとる方法と結合する方法の 2 つがあり，特に和を求める場合には以下のような Gate 機構 g_t^i を導入して重み付き和をとることが多い．

$$\begin{aligned} c_t^i &= g_t^i \odot c_{src,n}^{i-1} + (1 - g_t^i) \odot c_{trg,n}^{i-1} \\ g_t^i &= \sigma(W_g[c_{src,n}^{i-1}; c_{trg,n}^{i-1}]) \end{aligned}$$

ここで，記号 \odot はベクトルまたは行列の要素積を示す．この手法を用いる場合は新たな重み行列 $W_g \in \mathbb{R}^{H \times 2H}$ を必要とする．今回は他の Shared 型とパラメータ数を統一して実験を行うために Gate 機構は導入していない．

第 4 章 実験

4.1 データ

本研究で行う実験には、主に 14th International Workshop on Spoken Language Translation (IWSLT2017) の Bilingual Task*にて配布された対訳コーパス [16] を用いた。これは TED Talks†で公開されている動画に付与された字幕を元に作成したコーパスであるため、本論文では以下 TED コーパスと呼ぶ。複数の言語対のものが公開されているが、本研究では日英、独英、中英の 3 言語対を用いる。TED Talks の動画はそれぞれ 1 回のプレゼンテーションの様子を収めたものである。このコーパスでは各時刻に画面に表示される単語列を 1 系列としている。このため 2 文以上を含む場合もあるが、その場合も 1 系列を 1 文とみなして実験を行う。また、各動画がそれぞれ特有の文脈を有すると仮定し、動画 1 つ分の文集合を文書として扱う。評価データとしては 2010 年度から 2016 年度までのものが存在するが、今回は 2014 年度版を用いた。

日本語文には MeCab‡ (辞書に IPADic 2.7.0 を使用)、中国語文には jieba§, 英語文とドイツ語文には統計的機械翻訳のツールである Moses [17] に付属する `tokenizer.perl` をそれぞれ用いて単語分割した。その後、計算量等の都合から文長が 100 単語以上の文を含む文書を学習データからのみ削除した。

また、各文は Byte Pair Encoding (BPE) [18] という処理に通し、サブワード単位へ分割する。これは、ニューラル機械翻訳では使用語彙集合に含まれない単語を生成できないため、未知語や低頻度な単語を単語より小さなサブワードという単位にすることで、全ての単語を扱えるようにする処理である。具体的には、一度単語を文字単位に分割したのち、頻度の高い文字 2-gram を 1 つの文字として結合し、指定された結合回数だけ頻度の測定と結合を繰り返すという処理である。BPE の実装には subword-nmt¶を用いた。BPE の結合回数は 32,000 に設定した。

*<https://sites.google.com/site/iwslt2017/TED-tasks>

†<https://www.ted.com/talks>

‡<http://taku910.github.io/mecab/>

§<https://github.com/fxsjy/jieba>

¶<https://github.com/rsennrich/subword-nmt>

コーパス	学習	検証	評価	英語側の語彙数	他方の語彙数
TED 独英/英独	203,998	888	1,305	30,446	31,790
TED 中英/英中	226,196	879	1,297	30,571	37,928
TED 日英/英日	194,170	871	1,285	30,293	34,295
Recipe 日英/英日	108,990	3,303	2,804	7,886	11,103

表 4.1: 前処理を通して得られた各コーパスの統計量

また、ドメインの違いによる影響を調べるために、The 4th Workshop on Asian Translation (WAT2017) で配布された Recipe Corpus^{||}を用いた。これは、レシピを自由に投稿できる Web サイトである Cookpad^{**}に日本語で投稿されたものを英訳したデータである。このコーパスではレシピ中の各手順をそれぞれ 1 文とみなす。したがって、1 文とされた同一手順内に 2 文以上含まれている場合でも TED コーパスと同様に 1 文とみなして実験を行う。また、各レシピをそれぞれ 1 文書として扱う。前処理は TED コーパスと同様に行い、BPE の結合回数を 8,000 にして適用した。以上の処理で得られた各コーパスの統計量を表 4.1 に示す。

4.2 ニューラル機械翻訳の設定

本研究のベースラインは、Luong ら [3] の Dot Global Attention 型ニューラル機械翻訳である。単語分散表現と隠れ状態ベクトルの次元数は 512 に設定した。最適化には、AdaGrad を初期学習率を 0.01 に設定して用いた。Dropout の確率は 0.2 とした。評価時のみビーム探索を窓幅を 5 にして用いた。Encoder は 2 層の双方向 LSTM, Decoder 側は 2 層の単方向 LSTM をそれぞれ用いた。ミニバッチの大きさは 128 文書である。128 文書ごとに同じ文番号を持つ文が 1 つのミニバッチになるようにミニバッチ集合を作成し、文番号の昇順に処理を行った。以上の実験設定は、提案手法の実験でも同様である。

Separated 型の文脈 Encoder は、ベースラインと同様に 2 層の双方向 LSTM を

^{||} <http://lotus.kuee.kyoto-u.ac.jp/WAT/recipe-corpus/>

^{**} <https://cookpad.com/>

実験	Baseline	Separated 型		Source	Shared 型		Mix
		Source	Target		Target		
TED 独英	26.55	26.29 ± .37	26.52 ± .12	*27.20 ± .11	* 27.34 ± .11	27.18 ± .21	
TED 英独	21.26	21.04 ± .64	20.77 ± .10	21.63 ± .27	21.83 ± .30	21.50 ± .29	
TED 中英	12.54	12.52 ± .33	12.63 ± .24	*13.36 ± .41	* 13.52 ± .10	*13.23 ± .09	
TED 英中	8.97	8.94 ± .11	8.71 ± .06	9.45 ± .22	* 9.58 ± .13	9.42 ± .19	
TED 日英	5.84	*6.64 ± .26	*6.37 ± .12	*6.95 ± .07	* 6.96 ± .18	*6.81 ± .16	
TED 英日	8.40	8.58 ± .12	8.26 ± .00	8.51 ± .31	8.59 ± .08	8.66 ± .14	
Recipe 日英	25.34	*26.51 ± .09	*26.69 ± .15	*26.90 ± .17	* 26.92 ± .10	*26.78 ± .11	
Recipe 英日	20.81	*21.87 ± .12	*21.45 ± .14	* 22.02 ± .20	*21.97 ± .09	*21.81 ± .15	

表 4.2: 各実験の BLEU スコア

用いた。Separated Target 型の学習時は目的言語側の文脈文として出力結果ではなく正解データを使用し、検証・評価時には出力結果を与えた。Shared Target 型の学習時には検証・評価時と同様に文脈文の翻訳時に計算した状態ベクトルを用いるが、Decoder の入力として正解データを用いるため、正解データを元に計算された状態ベクトルである点が異なる。提案手法に存在する重み行列のうちベースラインにも存在する重み行列は、ベースラインモデルを事前学習した結果を初期値として与える。文書の先頭の文を訳す場合は文脈文が存在しないため、提案手法で導入する文脈 Attention \mathbf{c}^{i-1} はゼロベクトルとした。

評価には BLEU [19] を用いた。学習は各 30 Epoch 行い、検証データの BLEU スコアが最も高かった Epoch のモデルを評価に用いた。今回の実験では、ニューラルネットワークの重みの初期値や Dropout による初期化などの処理にランダム性を含む。このため、提案手法の有意性を調べるためにベースライン以外の全ての実験で乱数のシードを変えて 3 回ずつ実験を行い、その平均値と標準偏差を報告する。また、Travatar [20] に付属する bootstrapping resampling のツールによって、ベースラインの結果とその他の実験の間の統計的有意差を測定する。

4.3 結果

表 4.2 に結果を示す．なお，その言語対で最も性能が高かった値については太字で表記した．* は 3 回の実験とも統計的有意差 ($p < 0.05$) があったことを示す．

ベースラインと比較して全ての実験で Shared Target 型の性能が向上していることがわかる．中英翻訳や日英翻訳などの一部の実験では Separated Target 型でも向上しているが，Shared Target 型の結果と比べると低い値である．したがって，目的言語側の文脈文の情報は文脈 Encoder から取り込むのではなく，Decoder の計算履歴から取り込む方がよいと言える．

Separated Source 型と Shared Source 型はどちらも Encoder から情報を得ているため，結果に差異は生じないと考えていた．翻訳対象文を読み込む Encoder と文脈 Encoder の重み共有を行った Voita ら [11] の研究では，英露翻訳の実験でどちらも同程度の性能であったと報告している．しかし本研究で行った実験では，ほぼ全ての実験で Separated Source 型より Shared Source 型の性能の方が高かった．したがって，どちら側の文脈文を用いるとしても，Shared 型を用いることで性能を落とさずに計算コストだけを抑えた Multi-Encoder 型の文脈つきニューラル機械翻訳が得られると言える．Voita らの研究と本研究では，使用したデータの言語対や Transformer を用いているかどうかなどが異なっている．本研究の提案手法を Transformer を用いて再実験した際に傾向の差異が生じるかどうかについては，今後の課題として残したい．

第 5 章 考察

本章では，得られた実験結果を言語対の影響，学習の収束，重み共有による影響の 3 つの観点から考察する．また，5.4 節では出力例を用いた考察を行う．

5.1 言語対の影響

BLEU スコアは，言語対に大きく依存するものであった．例えば日英・英日翻訳では TED・Recipe とともに，Shared Source 型と Shared Target 型の差が小さいため，日英・英日翻訳ではどちら側の情報も同程度に必要であると言える．日本語文は省略される情報が多いため，日英翻訳では文脈を考慮することで足りない情報を補うことができ，英日翻訳では重複する情報を削ることでより流暢な訳出ができる．語族は違うがともに SVO 型の言語対である中英・英中翻訳ではどちらも目的言語側が有用であったため，語順や省略の有無などの言語的な特徴の違いが有用性に違いを生んでいると考えられる．

Shared Mix 型の実験結果は，ほとんどの言語対で Shared Source 型と同程度か少し低い水準であった．TED 英日翻訳では最高性能を達成していたが，Shared Target 型の結果に比べて有意に高い結果であるとは言えない．この結果から，Shared Mix 型で導入した Attention の和を用いる方法によってそれぞれの文脈情報が扱えていると仮定すれば，両方の文脈情報を合わせて用いる必要は薄く，言語対に応じてどちらかの文脈文を用いればよいということが言える．

Separated 型の実験では，いくつかの言語対でベースラインより低い結果となった．Recipe コーパスでは低下が見られなかったため，TED コーパスに固有のデータ特性であると思われる．また，Separated 型の結果から，先行研究で述べられていたように原言語側の文脈がより有用なのではなく，英語側の文脈が有用である可能性を指摘する．

5.2 学習の収束

図 5.1 に、各言語対の 1 回目の学習時における BLEU スコアの推移を示す。Shared 型や Separated Source 型の学習はとても安定していると言える。Shared 型は文脈 Encoder を学習する必要がなく、事前学習によって学習初期から一定の質が担保された隠れ状態ベクトルを用いることができる。Separated Source 型は文脈 Encoder を学習する必要があるが、今回の実験では、学習時も検証・評価時も同程度の質の入力文を使うことができる。したがって、学習の収束の観点から考えると、従来手法の拡張である Separated Source 型と提案手法である Shared 型は扱いやすい手法であると言える。

一方、Separated Target 型の学習は不安定であり、特に目的言語が英語でない場合に顕著になるように思われる。Separated Target 型の学習時には正解の文脈文を与えられているが、検証・評価時には実際の出力結果が文脈文として与えられている。このため、Decoder が目的言語らしい文を生成できるようになる前に文脈 Encoder が目的言語文の文法を学習してしまい、正しく読み込める Encoder で間違った文を読み込んだ結果、有用な文脈情報にならなかったのではないかと考えられる。したがって、Separated Target 型では生成しにくいかつ読み込みやすい言語を目的言語にした場合に学習が収束しにくい可能性がある。今回の実験結果から、英語がそのような言語に該当すると考えられる。

5.3 重み共有の影響

ニューラル機械翻訳では、Encoder と Decoder をそれぞれ多層にすることによって性能が向上することが知られている。Dabre と Fujita [21] は、 n 層を用いる際に n 個の重み行列を用意するのではなく、1 つの重み行列で再帰的に n 回計算しても性能が低下しないことを発見した。また著者らは、学習時と評価時で層の数を変える実験を通して、重み行列が学習時の層数を記憶し、層ごとの役割を模倣していることを発見した。これは、層方向の重み行列を共有できることを示している。本研究の提案手法である Shared 型も重み行列を共有しているという観点は共通している。Encoder または Decoder を 1 層目、文脈 Encoder を 2 層目と考えることによって、Dabre と Fujita の研究を文書の時系列方向に対して拡張したものとして扱

える可能性がある。特に，Shared Source 型が Separated Source 型と比較してパラメータ数が少ないにもかかわらずより高性能であったことから，重み行列を共有することによってニューラル機械翻訳の空間を効果的に扱えていると考えられる。

また本研究は，複数のタスクを 1 つの共有されたニューラルネットワークで解くことで性能を高める，マルチタスク学習の枠組みでも捉えることができる。提案手法を機械翻訳と別のタスクとのマルチタスク学習と捉えたとき，別のタスクに当たるものは QuickThought [22] が近いといえる。これは，ある文脈文が与えられたとき，それに続く文として適切な文を選択するタスクを解くことで質の良い文ベクトルを得るものである。したがって，本研究は文脈を扱うことで性能向上を図るものであったが，実験結果はそれに加えてマルチタスク学習による性能向上の影響も受けている可能性がある。

5.4 出力例

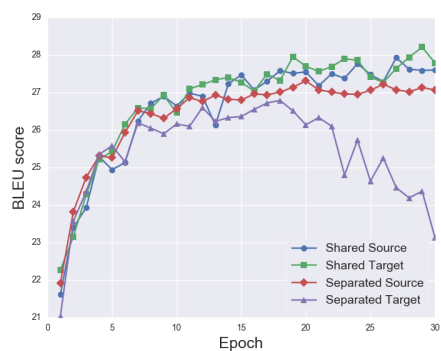
本節では，提案手法が目的言語側の文脈を扱えているかどうかについて議論するため，出力文の訳語の一貫性に着目した考察を行う。表 5.1 に Recipe 日英翻訳の出力例を示す。それぞれの上段と下段は連続した 2 文である。出力結果については，この例の 1 文目（各上段）は文書の 1 文目であるため，どの手法であっても文脈を用いない場合の結果である。2 文目（各下段）については，Separated Source 型と Shard Source 型は文脈文として 1 文前の入力文を与え，Separated Target 型と Shared Target 型は文脈文として 1 文前の出力文（上段の文）を与えた結果である。Shared Mix 型については 1 文前の入出力文を文脈文として与えた。

“わかめ”と“長ねぎ”についてベースラインの出力例を見ると，1 文目では“wakame seaweed”と“Japanese leek”に訳されているのに対し，2 文目ではそれぞれ“wakame”と“leek”に訳されている。Separated Source 型と Shared Source 型でも，“長ねぎ”についてはベースラインと同様であった。また，Separated Target 型では Separated Source 型の特徴に加えて，1 文目の“塩を落とし”に当たるものが不足している。いずれの場合も，2 文目のみについて考えると妥当性*が高い。また，該当箇所に関しては参照訳と同じフレーズを生成できていることから，

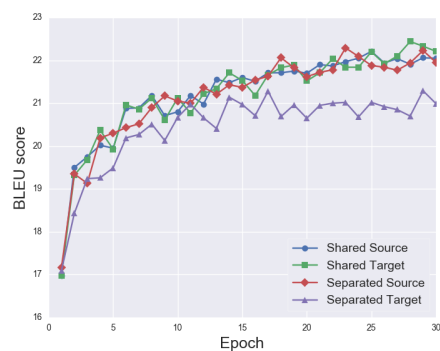
*翻訳の妥当性とは，訳文の質を入力文と出力文の間での文意の過不足について評価する観点である。

BLEU スコアも高いと言える。しかし、1 文目も考慮しながら俯瞰で評価をした場合、これらの生成結果は訳語の一貫性に欠ける。

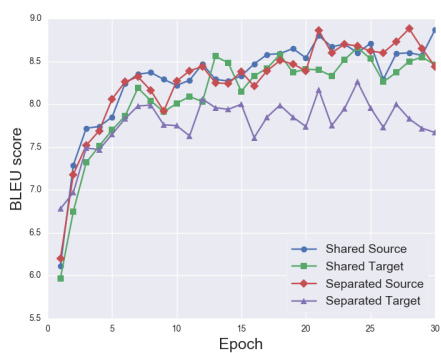
一方で、Shared Target 型と Shared Mix 型では、1 文目・2 文目ともに “wakame” と “Japanese leek” に訳されている。一般に、訳語の一貫性を考えるためには過去の翻訳結果を参照しなければならないため、これらの提案手法により文書内の訳語の一貫性を向上させることができたと言える。したがって、重み共有によって目的言語側の文脈情報をより効果的に扱えていると考えられる。



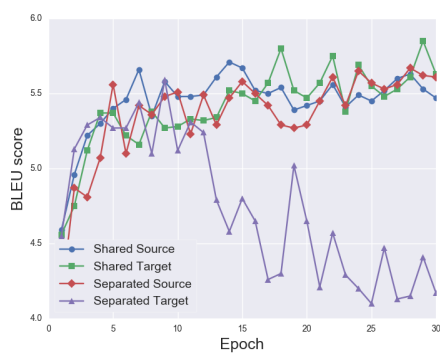
(a) TED 独英翻訳



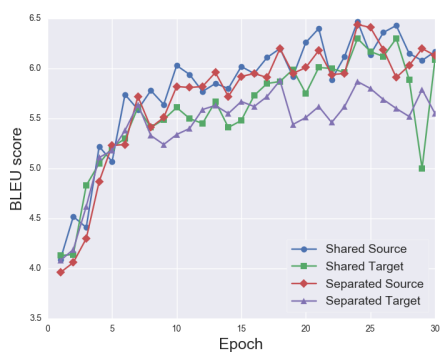
(b) TED 英独翻訳



(c) TED 中英翻訳



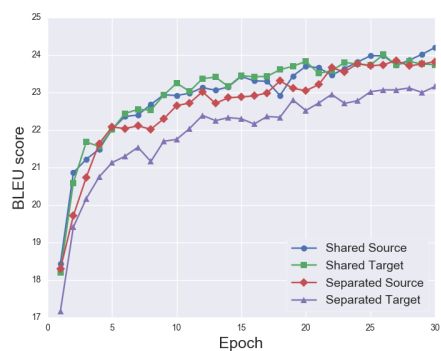
(d) TED 英中翻訳



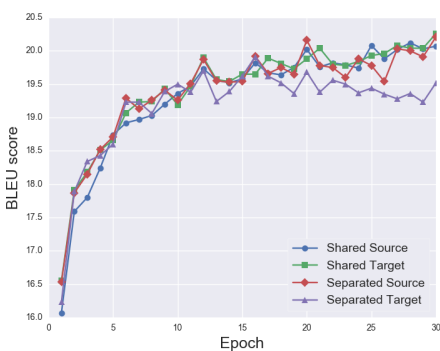
(e) TED 日英翻訳



(f) TED 英日翻訳



(g) Recipe 日英翻訳



(h) Recipe 英日翻訳

図 5.1: 各実験の学習時における検証データの BLEU スコアの変化

実験	文（後処理をかけた結果を記載）
入力文	わかめはよく洗って塩を落とし、10 分ほど水に浸けておいてからざく切りにする。長ねぎは小口切りにする。
	熱した鍋にごま油をひき、わかめと長ねぎを入れて 30 秒ほど軽く炒める。
参照訳	Wash the wakame well to remove the salt, put into a bowl of water for 10 minutes and drain. Cut into large pieces. Slice the Japanese leek .
	Heat a pan and pour the sesame oil. Stir fry the wakame and leek for 30 seconds.
Baseline	Wash the wakame seaweed well and remove the salt. Soak in water for 10 minutes, then roughly chop. Cut the Japanese leek into small pieces.
	Heat sesame oil in a heated pot, add the wakame and leek , and lightly sauté for about 30 seconds.
Separated Source	Wash the wakame well, remove the salt, soak in water for about 10 minutes, then roughly chop. Cut the Japanese leek into small pieces.
	Heat sesame oil in a heated pot and add the wakame and leek . Stir-fry for about 30 seconds.
Shared Source	Wash the wakame well, remove the salt, soak in water for about 10 minutes, then roughly chop. Cut the Japanese leek into small pieces.
	Heat sesame oil in a heated pot and add the wakame and leek . Stir-fry for about 30 seconds.
Separated Target	Wash the wakame well, soak in water for about 10 minutes. Cut into small pieces. Cut the Japanese leek into small pieces.
	Heat the sesame oil in a frying pan, add the wakame and leek , and stir-fry for about 30 seconds.
Shared Target	Wash the wakame well, remove the salt, soak in water for about 10 minutes, then roughly chop. Chop the Japanese leek into small pieces.
	Heat sesame oil in a heated pan, add the wakame and Japanese leek , and lightly stir-fry for about 30 seconds.
Shared Mix	Wash the wakame well, remove the salt, soak in water for about 10 minutes, then roughly chop. Chop the Japanese leek into small pieces.
	Heat sesame oil in a heated pan, add the wakame and Japanese leek , and stir-fry for about 30 seconds.

表 5.1: Recipe 日英翻訳の出力例

第 6 章 おわりに

本研究では，Multi-Encoder を用いた文脈つきニューラル機械翻訳における目的言語側の文脈文の有用性について調査した．先行研究の手法では目的言語の情報を Encoder を通して扱っていたが，本研究では Decoder を通して扱うことでさらに有用な情報となると考え，実験を行った．6 言語対の実験の全てで，目的言語側の文脈文を用いても原言語側の文脈文と同程度かそれ以上の有用性を得られることを示した．どちらの文脈がより有用であるかについては言語対やデータに依存することがわかったが，本実験で使用したデータでは，英語側の文脈を取り込むことが性能向上につながっている可能性が高かった．また，Shared Source 型の実験結果から，重み行列を共有すること自体が性能を引き上げる可能性も示唆した．

本研究は，より低資源な計算環境で文脈つきニューラル機械翻訳を実現できる可能性を示した．提案手法が文書単位の機械翻訳の実現に向けた研究開発の一助となることを願っている．

発表リスト

査読あり

1. Satoru Katsumata, Yukio Matsumura, Hayahide Yamagishi and Mamoru Komachi. **Graph-based Filtering of Out-of-Vocabulary Words for Encoder-Decoder Model**. In Proceedings of ACL 2018 Student Research Workshop, pp.112-119. Melbourne, Australia. July 17, 2018.
2. Michiki Kurosawa, Yukio Matsumura, Hayahide Yamagishi and Mamoru Komachi. **Japanese Predicate Conjugation for Neural Machine Translation**. In Proceedings of NAACL 2018 Student Research Workshop, pp.100-105. New Orleans, Louisiana, USA. June 2, 2018.
3. Hayahide Yamagishi, Shin Kanouchi, Takayuki Sato, Mamoru Komachi. **Improving Japanese-to-English Neural Machine Translation by Voice Prediction**. In Proceedings of the 8th International Joint Conference on Natural Language Processing (IJCNLP 2017), pp.277-282. Taipei, Taiwan. November 28, 2017.

査読なし

1. 平澤寅庄, 松村雪桜, 山岸駿秀, 小町守. 画像から抽出した複数種の特徴量を組み込んだニューラル機械翻訳の検討. NLP 若手の会第 13 回シンポジウム. August 29, 2018.
2. 吉村綾馬, 松村雪桜, 山岸駿秀, 小町守. 機械翻訳の自動評価のための N-

- best を用いたマルチリファレンス作成手法の提案. NLP 若手の会第 13 回シンポジウム. August 27, 2018.
3. 勝又智, 松村雪桜, 山岸駿秀, 小町守. ニューラル機械翻訳における共起情報を考慮した語彙選択. 言語処理学会第 24 回年次大会, pp.1058-1061. March 15, 2018.
 4. 黒澤道希, 松村雪桜, 山岸駿秀, 小町守. 述語の活用情報を用いたニューラル日英翻訳. 言語処理学会第 24 回年次大会, pp.813-816. March 14, 2018.
 5. 勝又智, 松村雪桜, 山岸駿秀, 小町守. ニューラル日英翻訳における RNN モデルと CNN モデルの出力分析. NLP 若手の会 (YANS) 第 12 回シンポジウム. September 4, 2017.
 6. 黒澤道希, 山岸駿秀, 松村雪桜, 小町守. 活用情報を用いた日英ニューラル機械翻訳. NLP 若手の会 (YANS) 第 12 回シンポジウム. September 4, 2017.
 7. 嶋中宏希, 山岸駿秀, 松村雪桜, 小町守. クロスリンガルな単語分散表現を用いた機械翻訳自動評価手法の検討. NLP 若手の会 (YANS) 第 12 回シンポジウム. September 3, 2017.
 8. 山岸駿秀, 叶内晨, 佐藤貴之, 小町守. ニューラル日英翻訳における出力文の態制御. 言語処理学会第 23 回年次大会, pp.1065-1068. March 16, 2017.
 9. Hayahide Yamagishi, Shin Kanouchi, Takayuki Sato and Mamoru Komachi. **Controlling the Voice of a Sentence in Japanese-to-English Neural Machine Translation**. In Proceedings of The 3rd Workshop on Asian Translation (WAT): System description papers, pp.203-210. Osaka, Japan, December 12, 2016.
 10. 山岸駿秀, 佐藤貴之, 叶内晨, 小町守. ニューラル日英翻訳における態の制御. NLP 若手の会 (YANS) 第 11 回シンポジウム. August 30, 2016.

謝辞

2016 年 4 月の研究室配属から 3 年間，研究の指導をしていただきました小町守先生には深く感謝いたします．右も左もわからなかった学部生のころから何度も国際会議の場で発表する機会を作っていただいたことは，自分にとって大きな経験となりました．3 年間を通して，自分を信頼してさまざまなタスクを与えていただいたおかげで，大きく成長できたと思っています．ありがとうございました．

研究室の同期・先輩・後輩の方々や秘書の馬縹亜紀子さんにも感謝いたします．特に WAT の論文を添削していただいた叶内晨さん，NMT のいろはを教えていただいた佐藤貴之さんにはとてもお世話になりました．ありがとうございました．

また，学外の機械翻訳コミュニティの方々にも感謝しております．たくさんのコメントやアドバイスをいただき，どうもありがとうございました．

そして，本研究の副査を引き受けていただいた山口亨先生と高間康史先生に謝意を表します．ありがとうございました．

参考文献

- [1] I. Sutskever, O. Vinyals, and Q.V. Le, “Sequence to sequence learning with neural networks,” *Advances in Neural Information Processing Systems* 27, pp.3104–3112, 2014.
- [2] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *Proceedings of the International Conference on Learning Representations*, pp.1–15, 2015.
- [3] T. Luong, H. Pham, and C.D. Manning, “Effective approaches to attention-based neural machine translation,” *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp.1412–1421, 2015.
- [4] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y.N. Dauphin, “Convolutional sequence to sequence learning,” *Proceedings of the 34th International Conference on Machine Learning*, pp.1243–1252, 2017.
- [5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in Neural Information Processing Systems* 30, pp.5998–6008, 2017.
- [6] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol.9, no.8, pp.1735–1780, Nov. 1997.
- [7] M. Müller, A. Rios, E. Voita, and R. Sennrich, “A large-scale test set for the evaluation of context-aware pronoun translation in neural machine translation,” *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp.61–72, 2018.
- [8] J. Tiedemann and Y. Scherrer, “Neural machine translation with extended context,” *Proceedings of the Third Workshop on Discourse in Machine Translation*, pp.82–92, 2017.
- [9] S. Jean, S. Lauly, O. Firat, and K. Cho, “Does neural machine translation benefit from larger context?,” *CoRR*, vol.abs/1704.05135, pp.1–6, 2017.
- [10] R. Bawden, R. Sennrich, A. Birch, and B. Haddow, “Evaluating discourse phenomena in neural machine translation,” *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp.1304–1313, 2018.
- [11] E. Voita, P. Serdyukov, R. Sennrich, and I. Titov, “Context-aware neural machine translation learns anaphora resolution,” *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp.1264–1274, 2018.
- [12] J. Zhang, H. Luan, M. Sun, F. Zhai, J. Xu, M. Zhang, and Y. Liu, “Improving

- the Transformer translation model with document-level context,” Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp.533–542, 2018.
- [13] L. Wang, Z. Tu, A. Way, and Q. Liu, “Exploiting cross-sentence context for neural machine translation,” Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp.2826–2831, 2017.
 - [14] Z. Tu, Y. Liu, S. Shi, and T. Zhang, “Learning to remember translation history with a continuous cache,” Transactions of the Association for Computational Linguistics, vol.6, pp.407–420, 2018.
 - [15] S. Maruf and G. Haffari, “Document context neural machine translation with memory networks,” Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp.1275–1284, 2018.
 - [16] M. Cettolo, C. Girardi, and M. Federico, “WIT³: Web inventory of transcribed and translated talks,” Proceedings of the 16th Conference of the European Association for Machine Translation, pp.261–268, 2012.
 - [17] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, “Moses: Open source toolkit for statistical machine translation,” Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions, pp.177–180, 2007.
 - [18] R. Sennrich, B. Haddow, and A. Birch, “Neural machine translation of rare words with subword units,” Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp.1715–1725, 2016.
 - [19] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “BLEU: a method for automatic evaluation of machine translation,” Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp.311–318, 2002.
 - [20] G. Neubig, “Travatar: A forest-to-string machine translation engine based on tree transducers,” Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp.91–96, 2013.
 - [21] R. Dabre and A. Fujita, “Recurrent stacking of layers for compact neural machine translation models,” Proceedings of the 33rd AAAI Conference on Artificial Intelligence, pp.1–6, 2019.
 - [22] L. Logeswaran and H. Lee, “An efficient framework for learning sentence representations,” Proceedings of the International Conference on Learning Representations, pp.1–16, 2018.